Refine Search

Search Results -

Terms	Documents
L17 and galois	10

Database:
US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L18

Refine Search

Recall Text | Clear | Interrupt

Search History

DATE: Monday, September 27, 2004 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
DB=USPT; PLUR=YES; OP=OR			
L18	L17 and galois	10	L18
L17	L16 and l12	17	L17
L16	L13 same syndrome	43	L16
L15	L14 and l12	19	L15
L14	L13 same polynomial	360	L14
L13	linear adj1 feedback	2135	L13
L12	bch	2099	L12
L11	L10 same galois	0	L11
L10	calculat$ near2 l6	260	L10
L9	L8 and l1	1	L9
L8	root same polynomial	1095	L8
L7	L6 and l1	1	L7
L6	correction adj1 term	2387	L6
L5	L1 and operand	0	L5

L4	L1 and 2t	1	L4
L3	L2 and l1	0	L3
L2	correctable near2 error	2026	L2
L1	6571368[pn]	1	L1

END OF SEARCH HISTORY

First Hit Fwd Refs Previous Doc Next Doc Go to Doc#

Generate Collection Print

L33: Entry 1 of 5 File: USPT Jun 15, 2004

DOCUMENT-IDENTIFIER: US 6751743 B1
TITLE: Method and apparatus for selecting a first clock and second clock for first and second devices respectively from an up-converted clock and an aligned clock for synchronization

Detailed Description Text (22):
FIG. 6 illustrates a block diagram of portions of protocol processor 520 that includes a framer 600. Deserializer 650 produces a parallel signal 610 from an incoming SONET signal 601 (e.g., an OC-192 data stream) from line side optical receivers (not shown). Receive module 605 processes the parallel signal 610, optionally processes the forward error correction (FEC) information and de-interleaves the OC-192 signal into four OC-48 line rate signals 615 for delivery to downstream OC-48 processors. Transmit module 620 processes four incoming OC-48 system rate signals from the OC-48 processors (signals 625), optionally inserts forward error correction information, and interleaves the four OC-48 signals into an OC-192 signal 630 for transmission by line side optical transmitters (not shown). A CPU Interface module 635 provides the CPU connection to the internal device registers.

Previous Doc Next Doc Go to Doc#

L18: Entry 1 of 10 File: USPT Jan 27, 2004

DOCUMENT-IDENTIFIER: US 6684364 B2
TITLE: Forward error corrector

Brief Summary Text (7):
One class of error correcting codes is the well-known BCH (Bose-Chaudhuri-Hocquenghen) codes, which include the Reed-Solomon ("RS") code. The mathematical basis of the RS code is explained in, e.g., the aforementioned reference by Lin et al. and also in Berlekamp, "Algebraic Coding Theory", McGraw-Hill, 1968, which is further referred to in U.S. Pat. No. 4,162,480 issued to Berlekamp. The aforementioned references are hereby incorporated by reference in pertinent part.

Brief Summary Text (9):
The invention herein provides a method and apparatus for decoding an algebraic-coded message. The method can include the steps of determining a discrepancy indicator, with the discrepancy being between a calculated and a predicted value; determining an error locator polynomial using a selected class of error correction algorithms, such as, for example, a Berlekamp-Massey algorithm; and detecting an uncorrectable message using the selected error correction algorithm. The apparatus is composed of storage devices which can include recirculating storage devices; arithmetic components attached to the storage devices, the components operating over a Galois Field on selected contents of the storage devices; and an uncorrectable message detector, connected with the storage devices and the arithmetic components.

Detailed Description Text (2):
The invention herein provides an apparatus for and a method of decoding algebraic codes, including BCH codes, and more specifically, Reed-Solomon codes, such that uncorrectable messages, or portions of received encoded data, are detected. Furthermore, the invention herein provides for a more area-efficient device implementation of the aforementioned method. For the purposes of illustration, the present invention will be described in terms of a subset of the BCH codes, namely Reed-Solomon (RS) codes.

Detailed Description Text (3):
The Reed Solomon (RS) encoding technique appends to each block of k user data symbols 2t redundancy symbols to create an encoded message block (where t represents the designed symbol error correcting capacity of the code). These 2t symbols, or elements, are selected from the Galois Field to be the roots of the generator polynomial. Therefore, there are k+2t symbols in a RS-encoded message block. The entire message block is viewed as a polynomial and evaluated as a polynomial at some Galois Field element. The Galois Field element at which the polynomial is evaluated will be located at one roots of the generator polynomial that are used to create the RS code. The RS code views the n-bit symbols as elements of a Galois Field (GF(2.sup.n)). A Galois field is a finite field, the elements of which may be represented as polynomials in a, where a is a root of an irreducible polynomial of degree n. The RS codeword consists of a block of n-bit symbols. Typically, n=8 and the 8-bit symbols are referred to as bytes. Constructing the Galois field GF(2.sup.n) requires a defining polynomial F(x) of degree n. In addition, a primitive element .beta. is chosen so that every nonzero element of GF(2.sup.n) is a power of .beta.. The element .beta. is not necessarily

a root of F(x).

Detailed Description Text (8):
Two frequently-used RS error correction algorithms are the Berlekamp-Massey and the Euclid algorithms. The present invention recasts the Berlekamp-Massey algorithm such that the inversion process typically associated with the traditional Berlekamp-Massey (tBM) algorithm is eliminated. This is important because the inversion process includes determining the reciprocal of certain Galois field elements using division. Division is a time consuming arithmetic operation, the implementation of which can occupy needed component area in a device design. Therefore, the present invention can be particularly advantageous where area-efficient layout of a decoder device is desirable.

Detailed Description Text (9):
For further elaboration of the decoding process over Galois fields, including tBM, Chien searching, and Forney's Algorithm, see Theory and Practice of Error Control Codes by Richard E. Blahut (Addison-Wesley, 1983) which is incorporated by reference in pertinent part herein.

Detailed Description Text (21):
FIG. 2 exemplifies an embodiment of the process 20 implementing the aforementioned improved control structure in the context of the mBM algorithm recited in Equations 4, 5, and 6(a)-(b). Although the implementations described herein are postured for standard RS codes having a block length of, for example, 255 elements, such implementations also may be used in the context of extended Reed-Solomon codes which, in the example herein, would have 256 elements in the message block, i.e., have 256 elements in associated the Galois Field. It is desirable that, in step 21, the control variables DEG, PWR, and STATE, as well as error locator variables be initialized. It further is desirable to iterate through steps 23, 24, 25, and 26, 2t times, where 2t is the number of syndrome polynomials to be evaluated, and t is the error correcting capability of the preselected code. Thus, at step 30, a counter tracking the number of completed iterations is employed. No additions or subtractions are needed in implementing the control variables, and only count up or down functions are used. Step 23 essentially implements Equation 4, in which the discrepancy value DEL, associated with a particular iteration, is determined. Similarly, step 24 implements Equation 5 in which the error locator polynomial is updated. In step 25, auxiliary polynomial B.sub.i is updated according to Equation 6a in substep 27, or Equation 6b in substep 28, based on conditions determined by logic 26. For logic 29, it is desirable for both STATE=ALPHA AND DEL < > zero to direct the data flow via an implementation of Equation 6a in substep 27; otherwise substep 28 is used, implementing Equation 6b. Unlike the tBM algorithm where the polynomial shift term (1-.delta..sub.r)X in Equation 3 has been normalized, the mBM algorithm does not require normalization, avoiding an inversion/division operation. After the auxiliary polynomial is updated in step 25, the controller state is updated in step 26.

Detailed Description Text (22):
In general, the degree of the error locator polynomial is tracked by DEG, which is an upcounter descriptive of the true degree of the error locator polynomial and, thus, the number of errors in the message block. It also is desirable to construct an error locator polynomial who roots equate to the locations of an error. Essentially, process 20 attempts to synthesize a linear feedback shift register (LFSR) that predicts the values of the syndrome polynomial. Such a LFSR can be useful to find the error locations. Discrepancy value, DEL, then exposes a discrepancy between the predicted value of the syndrome polynomial, and the value of the current syndrome polynomial, and invites further processing to discover the location of the errors. PWR is a counter that keeps track of the number of times that the controller previously remained in STATE=ALPHA. It is desirable to have the STATE remain in control state BETA for a count equivalent to the number of times that STATE previously remained in control state ALPHA.

Detailed Description Text (24):
FIG. 3 is an exemplary embodiment of a polynomial solver using the mBM algorithm. Solver 50 can include syndrome generator register 51, recirculating syndrome register 52, first delay register 53, locator polynomial (.LAMBDA..sub.i) register 54, auxiliary polynomial (B.sub.i) register 55, second delay register 56, first multiplier 57, second multiplier 58, adder 59, .DELTA. register 60, and .DELTA._register 61. In another embodiment, syndrome generator 51 can be separate from solver 50. It is desirable for multipliers 57, 58, and adder 59 to operate on Galois Field elements. It also is desirable for register 51 to be logically arranged as a circular register or loop, such that particular register values can be used in a pre-defined sequence. Furthermore, it is desirable that registers 52, 54, and 55 be logically arranged as push-down stacks or FIFOs, and also that the values contained therein rotate synchronously. The error locator polynomial .LAMBDA..sub.i are arranged in register 54 such that the least significant coefficient is at the top and the stack "grows down" as subsequent values are determined. It is desirable for the syndrome recirculating register 52 to operate such that the least significant coefficient is aligned with the bottom of the register, and the most significant with the top.

Previous Doc Next Doc Go to Doc#

L15: Entry 7 of 19 File: USPT Sep 4, 2001

DOCUMENT-IDENTIFIER: US 6286123 B1
TITLE: Circuit for calculating error position polynomial at high speed

Brief Summary Text (5):
In digital communication and storage systems, a Reed-Solomon (hereinafter, referred to as the RS) code for controlling an error has widely been used. In data coded to the RS code, an error is frequently generated during data transmission or reproduction. Since incorrect data is received due to this error, the data coded into the RS code needs to be error-corrected by RS decoding. In this RS decoding process, it is necessary to calculate an error position polynomial having an error position as a root. Examples of such error position polynomial calculation are described in papers, R. E. Blahut, "Theory and Practice of Error Control Code", Addison-Wesley, 1983, and J. L. Massey, "Shift Register Synthesis and BCH Decoding" IEEE Transactions on Information Theory, Vol. IT-15, pp. 122-127, January, 1969. As a circuit using the Berlekamp-Massey algorithm (BMA) to calculate the error position, there is an example using a linear feedback shift register (LFSR) published by Massey, 1965. A discrepancy is calculated from a syndrome and an error position polynomial. If the discrepancy is 0, the previous error position polynomial is used. If it is not 0, the error position polynomial is again calculated. In order to again calculate the error position polynomial, a correction polynomial is given. That is, a new error position polynomial is calculated by using the correcting polynomial and the discrepancy. However, since such a circuit has a parallel structure, many multipliers are required to calculate the error position polynomial, and thus, the size of the circuit is increased. Moreover, since there is a long delay time in a circuit for calculating the discrepancy and a circuit for calculating the error position polynomial using the discrepancy, it is difficult to apply such a conventional circuit to a digital communication system which pursues a high-speed operation or to a storage system which has high storage capacity and requires high-speed access.

L15: Entry 2 of 19 File: USPT Jan 27, 2004

DOCUMENT-IDENTIFIER: US 6684364 B2
TITLE: Forward error corrector

Brief Summary Text (7):
One class of error correcting codes is the well-known BCH (Bose-Chaudhuri-Hocquenghen) codes, which include the Reed-Solomon ("RS") code. The mathematical basis of the RS code is explained in, e.g., the aforementioned reference by Lin et al. and also in Berlekamp, "Algebraic Coding Theory", McGraw-Hill, 1968, which is further referred to in U.S. Pat. No. 4,162,480 issued to Berlekamp. The aforementioned references are hereby incorporated by reference in pertinent part.

Detailed Description Text (2):
The invention herein provides an apparatus for and a method of decoding algebraic codes, including BCH codes, and more specifically, Reed-Solomon codes, such that uncorrectable messages, or portions of received encoded data, are detected. Furthermore, the invention herein provides for a more area-efficient device implementation of the aforementioned method. For the purposes of illustration, the present invention will be described in terms of a subset of the BCH codes, namely Reed-Solomon (RS) codes.

Detailed Description Text (22):
In general, the degree of the error locator polynomial is tracked by DEG, which is an upcounter descriptive of the true degree of the error locator polynomial and, thus, the number of errors in the message block. It also is desirable to construct an error locator polynomial who roots equate to the locations of an error. Essentially, process 20 attempts to synthesize a linear feedback shift register (LFSR) that predicts the values of the syndrome polynomial. Such a LFSR can be useful to find the error locations. Discrepancy value, DEL, then exposes a discrepancy between the predicted value of the syndrome polynomial, and the value of the current syndrome polynomial, and invites further processing to discover the location of the errors. PWR is a counter that keeps track of the number of times that the controller previously remained in STATE=ALPHA. It is desirable to have the STATE remain in control state BETA for a count equivalent to the number of times that STATE previously remained in control state ALPHA.